A Fuzzy Approach to Qualification in Design Exploration for Autonomous Robots and Systems

Jeremy Morse,
Dejanira Araiza-Illan,
and Kerstin Eder
Dept. of Computer Science
University of Bristol
Bristol, BS8 1UB, UK
Email: {jeremy.morse,dejanira.araizaillan,
kerstin.eder}@bristol.ac.uk

Jonathan Lawry
Dept. of Engineering Mathematics
University of Bristol
Bristol, BS8 1UB, UK
Email: j.lawry@bristol.ac.uk

Arthur Richards
Dept. of Aerospace Engineering
University of Bristol
Bristol, BS8 1TR, UK
Email: arthur.richards@bristol.ac.uk

***Abstract*—Autonomous robots must operate in complex and changing environments subject to requirements on their behaviour. Verifying absolute satisfaction (true or false) of these requirements is challenging. Instead, we analyse requirements that admit flexible degrees of satisfaction. We analyse vague requirements using fuzzy logic, and probabilistic requirements using model checking. The resulting analysis method provides a partial ordering of system designs, identifying trade-offs between different requirements in terms of the degrees to which they are satisfied. A case study involving a home care robot interacting with a human is used to demonstrate the approach.**

I. Introduction

If robots are to interact with people routinely, evidence must be provided that they are both safe and function as intended, i.e. useful. Verification by post-implementation experimenting is expensive, as is correcting any problems found at that stage. Hence, verification is preferred at design time, where many aspects of the system, its requirements, and the environment, might not be precisely defined and specified. We therefore propose a more flexible form of verification to assess levels of requirement satisfaction at design time, rather than strict pass or failure. In particular, we consider two different kinds of requirements: probabilistic requirements (PRs) that formally define requirements which hold with some measurable probability (e.g. the robot needs to reach a safe location in 30 seconds with a probability of 0.8); and vague requirements (VRs), i.e. requirements that can be specified as a scale between unacceptable and completely acceptable (e.g. the velocity of the robot should be preferably low to avoid dangerous collisions).

The contribution of this paper is a single analysis method for requirement satisfaction, combining the use of fuzzy logic for VRs, and measuring probabilities of satisfaction for PRs. These metrics provide a partial ordering of different design candidates that represents a flexible specification. Every design candidate is evaluated both in terms of fuzzy set membership for every VR and probability for every PR, calculated using probabilistic model checking [1]. Our analysis ensures the levels of VR and PR satisfaction for each candidate are directly comparable. The combination enables trade-offs between the different kinds of requirements to be investigated during the robot design activity, allowing ranking and discrimination between design candidates.

Consider an autonomous home-care assistant which must periodically attend its user. To avoid harmful collisions, it is required that the robot movement be 'slow', an example of a VR since the precise threshold of acceptability is to be determined. However, it is also required that the robot should reach the user quickly when called, considered as a PR due to the uncertain location of the user. Since the latter requires the robot to move quickly, it competes with the 'slow' VR.

The presence of multiple requirements has been identified as a serious research challenge for autonomous systems design, e.g. when dealing with robots and environments that adapt, due to associated uncertainty and vagueness [2]. The self-adaptive software research roadmap in [3] proposes that requirements be given in *flexible* terms, allowing a broad interpretation of when a system satisfies a requirement and when it does not. Thus, formalization and analysis methods for both PRs and VRs combined is timely and necessary to aid designers towards safe and functional robots.

An outline of the remainder of this paper is as follows: In Section II we give a brief overview of related work. Section III introduces concepts underpinning our work. Section IV formalizes a system specification. Section V proposes how fuzzy logic can be used to represent VRs. In Section VI we describe the care assistant case study and use it to illustrate and evaluate our approach. Finally, in Section VII we conclude and give an outlook on future research.

II. Related Work

There is considerable literature on the verification of requirements at design time, i.e. before system deployment in the real world. Available approaches for a well defined set of requirements include formal methods (e.g. model checking in [4], [5]), and simulation-based testing (e.g. as in [6]). The complexity of verifying autonomous systems at design time can be addressed by runtime verification, especially for systems with adaptation capabilities (e.g. [7], [8]). This still

leaves us with choices to consider at design time, such as the adaptation methodologies, or physical constraints due the choice of sensors and actuators.

Formal methods can be used to verify that PRs are satisfied at design time, e.g. probabilistic model checking [1]. Trade-offs in the simultaneous satisfaction of multiple PRs from a multi-objective perspective can be computed (i.e. a Pareto front) from solving games with rewards in probabilistic model checking [9]. From another perspective, many formal approaches have been proposed for correct-by-construction controller synthesis in robotics and autonomous systems (e.g. [10]). Strategies and controllers that partially satisfy a temporal logic property [11], or that violate a number of assumptions about a problem the least [12], can be synthesized. Overall, formal approaches at design time are effective for analysing PRs, but do not address VRs and, to date, have not been combined with VR techniques.

For VRs, the literature focuses on quantifying them through metrics (e.g. in multi-objective optimization used for design exploration [13]). In the field of requirements engineering, techniques have been proposed for reasoning about trade-offs between requirements that can only be partially satisfied at design time [14], and also techniques to model qualitative VRs through fuzzy logic [15], albeit at runtime. Requirement languages such as RELAX [16], [17] provide a means for describing and quantifying partial satisfaction of a VR. The problem of finding a system configuration that satisfies functional and desirability criteria (VRs) has been formalized by [18]. Existing techniques either limit the VRs that can be expressed (such as requiring independence [14]) or are not suitable for design time analysis. In addition, while most of the approaches offer formal descriptions of trade-offs with VRs, practical techniques to establish if a system satisfies all types of requirements (PRs and VRs) remains a challenge.

III. Preliminaries

For a completely deterministic system with states $x \in X$ and a transition relation $R \subseteq X \times X$, model checking tools can be used to determine if a requirement, formalized as a temporal logic formula ϕ, holds from a given start state $x_0 \in X$. More generally, we are often faced with systems with components that can behave randomly, e.g. human behaviour, and which are also non-deterministic or imprecisely defined, e.g. where there are only limited design-time assumptions about the robot controller. Such systems can be modeled as a Markov Decision Process (MDP).

Definition 1: A MDP (with labelling function) is a tuple M = (X, x_0, U, P, Π, L) where:

- X is a finite set of states;
- $x_0 \in X$ is the initial state;
- U is a finite set of actions;
- $P : X \times U \rightarrow Dist(X)$ is a transition probability function, which maps state-action pairs to a probability distribution over X $Dist(X)$;
- Π is a finite set of atomic propositions;
- $L : X \rightarrow 2^{\Pi}$ is a labeling function that assigns each state a set of atomic propositions in Π.

Temporal logic formulae over MDPs can be specified in PCTL [19] expressed over the labels of the MDP, which provides operators over probabilities to quantify the probability of a particular formula holding. The latter enables *quantitative verification*, where one can specify that a system will fulfil a PR with a certain probability. A formula can be evaluated over the system model to determine the precise probability with which it will hold. Such assurances, and thus quantitative verification and PRs, are valuable where requirements still "must" hold, but where external disturbances may cause them to be violated, such as uncontrolled environments. An autonomous vehicle, for example, may be capable of fulfilling its requirements, but cannot avoid a violation if another vehicle makes a serious error.

IV. Specifications

We now introduce our idea of a specification, to describe variations of system designs, and how existing techniques can be used to determine the level of PR satisfaction of a specification.

A system is defined by its *state* $x \in X$, changed by the enactment of an *action* $u \in U$, at each state. A *specification* is a two-valued membership function defined over states and actions. It delimits the actions that are allowed in each one of the states, always allowing at least one action per state, to avoid deadlock. Here we are referring to design constraints such as maximum robot speed, minimum battery charge, time to service the human etc.

Definition 2: A specification is a function of the state variables and the actions, $f : X \times U \rightarrow \{0, 1\}$, where $f(x, u) = 1$ defines an action u that is allowed in the state x.

Different system design constraints lead to a set of specifications, F, to analyse at design time. There is a natural sense of desirability in which specifications can be weakened or strengthened by, for instance, weakening or strengthening individual design constraints. This can be formalized to define a partial ordering on specifications as follows.

Definition 3: Given two specifications f and f', we say that f is a *weakening* of f', denoted $f \preceq f'$, if and only if $\forall x, u \in X \times U,\ f(x, u) \geq f'(x, u)$.

The relation $\geq$ is defined over the two-valued membership set $\{0, 1\}$. Notice that for a specification f, $f^{-1}(1)$ corresponds to the permissible set of pairs of states and actions. Clearly, $f \preceq f'$ if and only if $f^{-1}(1) \supseteq (f')^{-1}(1)$.

A. Model Checking a Specification

We produce a system representing that specification. Given an existing system described as an MDP (X, x_0, U, P, Π, L), we create a restricted system (X', x_0, U, P', Π, L), that conforms to a specification f. Here,

$$\begin{array}{lcll} P'(x,u) & = & P(x,u) & if\ f(x,u) = 1 \\ P'(x,u) & = & \emptyset & if\ f(x,u) = 0 \end{array} \quad (1)$$

where $\emptyset$ is the empty probability distribution, and X' is the subset of X that is reachable under the transition function P'.

Probabilistic model checking tools such as PRISM [1] can be used to determine the probability of a PR ϕ holding over an MDP as the system evolves, given the states and actions. In the presence of non-determinism (i.e. multiple actions per state) PRISM can evaluate any Markovian *policy* that resolves multiple actions to one per state. The evaluation of all policies forms a credal set $\mathcal{P}_f$, for a given specification f. From this we can determine lower and upper probabilities for ϕ, from:

$$\begin{aligned} \underline{P}_f(\phi) &= \max\{P(\phi) : P \in \mathcal{P}_f\} \\ \overline{P}_f(\phi) &= \min\{P(\phi) : P \in \mathcal{P}_f\}, \end{aligned} \tag{2}$$

where $P(\phi)$ is a single probability value for PR ϕ. Furthermore, for reasons that are beyond the scope of this paper, it also holds that for probabilistic model checkers if $f \preceq f'$ then $\mathcal{P}_f \supseteq \mathcal{P}_{f'}$. Consequently, the upper probability of the PR ϕ increases as the specification is weakened; i.e. if $f \preceq f'$ then $\overline{P}_f(\phi) \geq \overline{P}_{f'}(\phi)$. In the context of design-time verification, upper probabilities are also interesting since they are associated with the use of an optimal controller. In contrast, lower probabilities are associated with rather unrealistic optimal feedback controllers, such a robot that constantly moves away from its target. For the rest of this paper, we focus only on the maximum achievable success probability, corresponding to the best possible controller policies.

V. Fuzzy Modelling of Vaguely Defined Requirements

We propose a formalization of the level of satisfaction of VRs, i.e. informal requirements that range from unacceptable to fully acceptable, based on fuzzy logic. We apply a similar modeling approach to utility theory for multi-agent systems, where utilities are modeled in terms of fuzzy sets and membership functions (e.g. in [20]). Fuzzy logic allows flexible reasoning with imprecision and uncertainty, also helping to model vagueness of linguistic or intuitive information. We propose that VRs be modeled by normalised fuzzy sets on system actions as follows:

Definition 4: A vague requirement χ is defined as a fuzzy set with membership function $\mu_\chi : f^{-1}(1) \to [0, 1]$ such that $\sup\{\mu_\chi(x, u) : (x, u) \in f^{-1}(1)\} = 1$.

This can then be extended to specifications by quantifying the minimum level to when those state action pairs which satisfy a specification also specify χ.

Definition 5: Let f be a specification, $\mu_\chi(f)$ quantifies the level to which the specification f satisfies the VR χ, given by $\mu_\chi(f) = \inf\{\mu_\chi(x, u) : (x, u) \in f^{-1}(1)\}$.

Notice that, given this definition and Definition 3, it follows that if $f \preceq f'$ then $\mu_\chi(f) \leq \mu_\chi(f')$.

For separate VRs relating to different aspects of the system, a conjunction of these can be defined as $\chi = \chi_1 \wedge \ldots \wedge \chi_m$, where each VR χ_i is modeled as a fuzzy set. In this case the membership function μ_χ is determined using a t-norm T in the standard way so that $\forall(x, u) \in f^{-1}(1), \mu_\chi(x, u) = T(\mu_{\chi_1}(x, u), \ldots, \mu_{\chi_m}(x, u))$.

A. A Partially Ordered Set of Specifications

It is proposed that a finite set of specifications F are considered so as to provide reasonable coverage of the design space. Taken together with the weakening ordering in Definition 3 these generate a partially ordered set $(F, \preceq)$ which can be explored on the basis of levels of VR satisfaction. Moreover, the poset can also be explored on the basis of quantified satisfiability of PRs.

B. Complementary Use of PR and VR Analyses

We now bring together the PR evaluation from Section IV-A and our proposed formalization for VR partial satisfaction. In order to determine the specification with the highest level of VR satisfaction, amongst those which satisfy the PR ϕ to an appropriate probability threshold of ρ (according to quantitative model checking results), we need only consider the least weak of those specifications which satisfy ϕ. Given PR ϕ and VRs χ, we aim to identify

$$\arg\max\{\mu_\chi(f) : \overline{P}_f(\phi) \geq \rho,\ f \in F\}$$

for a specified satisfiability parameter $\rho \in [0, 1]$. Since $\overline{P}_f(\phi)$ increases and $\mu_\chi(f)$ decreases as f is weakened, it follows that this is equivalent to

$$\arg\max\{\mu_\chi(f) : f \in W\}$$

where $W = \{f \in F : \overline{P}_f(\phi) \geq \rho\}$.

The set W can be automatically identified by using search methods over a fully partial ordered set $(F, \preceq)$, although this might be computationally expensive. Alternatively, we could sample over F and use the acquired information to find these requirement "optimal" specifications at a more efficient computational cost (as is done in Gaussian process regression [13]). Overall, the complementary use of fuzzy logic and probabilistic model checking can allow the designer to observe the trade-off between satisfying the different VRs to a high level and satisfying the PRs (e.g. to a high probability), for system design exploration.

VI. Case study

A robotic home healthcare assistant needs to assist a human at certain times. The living space is laid out as a two-dimensional grid featuring a recharging point, visiting which results in the battery being instantly recharged. The human's behaviour is assumed to be random, as they occasionally run away from the robot.

A. Modeling of vague requirements

There are three safety design constraints; these are the maximum velocity limit v, the minimum energy margin before recharge e, and the maximum servicing time limit t. Thus, an action $u \in U$ is delimited by a velocity constraint, a battery charge constraint, and a servicing time limit. In our model, the design constraints to explore are defined as the sets of ordered constraints $t_\gamma = t \leq \gamma$ with $\gamma = 1, \ldots, 10$ units; $v_\delta = v \leq \delta$ with $\delta = 1, \ldots, 6$ units; and $e_\zeta = e \leq \zeta$ with $\zeta = 1, \ldots, 5, 10, 15, 20, 25$ units, respectively for the time, velocity

and energy limits. By taking conjunctions of these constraints we obtain a set of specifications $F = \{f_{\gamma,\delta,\zeta} : \gamma, \delta, \zeta\}$ where

$$f_{\gamma,\delta,\zeta} = t_\gamma \wedge v_\delta \wedge e_\zeta \text{ where } \gamma \in \{1,\ldots,10\}, \quad (3)$$
$$\delta \in \{1,\ldots,6\} \text{ and } \zeta \in \{1,\ldots,5,10,15,20,25\}.$$

Hence, the number of varied design constraints is $10+6+9=25$, and $|F| = 10 \times 6 \times 9 = 540$.

Our VRs are as follows, we wish the robot to move at a speed that minimizes the risk of a harmful collision, but to be able to reach the human quickly, while maintaining as low an energy threshold as possible before returning to the charging station. These are modeled by three fuzzy sets χ_1, χ_2 and χ_3. The membership functions of these fuzzy sets are shown in Fig. 1. The overall VR for the system is the conjunction $\chi = \chi_1 \wedge \chi_2 \wedge \chi_3$, where the membership for χ is determined by applying the $\min$ t-norm:

$$\mu_\chi(f) = \min(\mu_{\chi_1}(f), \mu_{\chi_2}(f), \mu_{\chi_3}(f))$$

The choice of membership functions capture the designer's subjective judgement concerning partial requirement satisfaction. For χ_1, we employ a point system for risk analysis with values from 1 to 25, where a low velocity has a lower collision risk (thus a smaller risk value) whereas a high velocity has a higher collision risk (thus a high risk value). Two possible membership functions are suggested for χ_1. According to the 'Sigmoid' function, a maximum velocity limit below 3 units, with an associated collision risk of below 5 units, is highly desirable; and a maximum velocity limit of 5 units is very undesirable, with a risk of above 10. In contrast, according to the 'Linear' membership function there is more tolerance of collision risk values between 5 and 15, with a steady decay in desirability. For χ_2, we proposed three different functions that represent a range of service time tolerances, from a robot that has to reach the human as quickly as possible at all times ('Very Fast'), to a robot that can choose to act as fast as possible or slightly slower ('Medium') according to the occasion. For χ_3, we only proposed one membership function, where an energy threshold of 3 or less is highly desirable, and above 8 is highly undesirable.

B. Model Checking of a probabilistic requirement

To evaluate our PR, we build an MDP model and evaluate it with the probabilistic model checker PRISM [1]. The PR ϕ is expressed as a PCTL [19] formula:

$$\phi = \neg service \ \mathsf{U} \ service \quad (4)$$

where $service$ is a label that is true for a state where the user is being attended, and false otherwise. Thus, our PR is that from any start state, the system is able to attend to the user with as high a probability as possible.

Our model is written in the PRISM modeling language, consisting of 5 modules, corresponding to the human and robot motion, the timing, the energy in the battery, and the servicing task. The movement of the human is a probabilistic choice of moving (north, south, east, west), or to stand still, each with a probability of 0.2. The robot and its control system are modeled as a set of non-deterministic choices. The robot may chose to move at a range of speeds in any of the four directions, or stay still. We leave the model checker to derive the best navigation policy. The robot has a battery with a finite amount of energy that, when fully discharged, causes the robot to stop. We filter out unrealistic robot behaviours, such as moving outside the grid environment, and motions that cause immediate collision with the human.

For each specification $f \in F$ we restrict the model as in Section IV-A by conjoining propositions with ϕ, for example $velocity \leq 5$, forcing the model checker to not explore any transitions where $velocity$ exceeds 5. Different time bounds in the specification are modeled by using the $\mathsf{U} \leq \alpha$ operator instead of U, specifying that ϕ must hold within α steps. The model checker is used to compute the maximum probability the design can achieve when operating under the specification, assuming an optimal controller, for every start state of the model (i.e. combinations of robot, human, battery life etc.).

C. Experimental Results

Experiments were run on a PC with Intel i5-3230M 2.60 GHz CPU, 8 GB of RAM, Ubuntu 14.04. We ran the model checking analysis phase on PRISM 4.2.beta1, for each of the 540 specifications. Model checking took less than 1 minute for each run, with a minimal computation time of 10 seconds. All underlying data and models are openly available online[1].

We computed the upper probabilities of satisfying the PR ϕ, with the system model constrained by each of the specifications $f \in F$ as explained in Section VI-B. Fig. 2 shows a Hasse diagram of the resulting probabilities for a subset of the 540 specifications of the form $f_{\gamma,\delta,5} = t_\gamma \wedge v_\delta \wedge e_5$, where $\gamma \in \{1,\ldots,10\}$, $\delta \in \{1,\ldots,6\}$. Fig. 2 also shows the upper probabilities $\overline{P}_f(\phi)$. As described in Section IV-A these are monotonically increasing when specifications are weakened according to Definition 3.

The probability bounds provided by $\overline{P}_f(\phi) \geq \rho$ form surfaces analogous to Pareto frontiers in multi-objective optimization, representing the "optimal" specifications in terms of meeting the probability threshold ρ for the property ϕ. For example, the strongest specifications such that $\rho = 0.9$ have been highlighted in Fig. 2.

We computed the membership value $\mu_\chi(f)$ for each specification, to quantify the level of VR partial satisfaction. Fig. 3 shows an example of this computation, for the same specifications as Fig. 2, using the 'Sigmoid' function as μ_{χ_1} and the 'Very Fast' function for μ_{χ_2} from Fig. 1. The results in Fig. 3 show $\mu_\chi(f_{\gamma,\delta,\zeta})$, in the same order as Fig. 2. We have highlighted the specifications with a probability $\overline{P}_f(\phi) \geq 0.9$ as in Fig. 2. The specifications that satisfy $\arg\max\{\mu_\chi(f) : f \in W\}$ are shown in blue. Fig. 4 shows a compressed version of the $10 \times 6 \times 9$ Hasse diagram with the strongest specification as the top element.

[1]Available from https://github.com/riveras/homecare

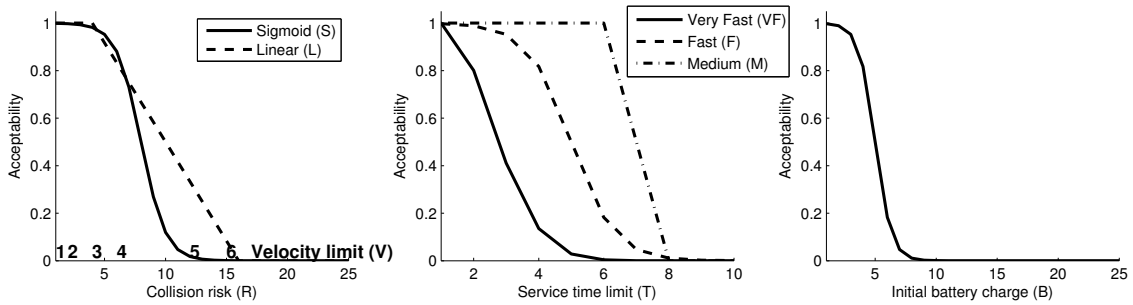

Fig. 1. Proposed membership functions to quantify partial VR satisfaction of system specifications, with respect to the VRs χ_1, χ_2 and χ_3.

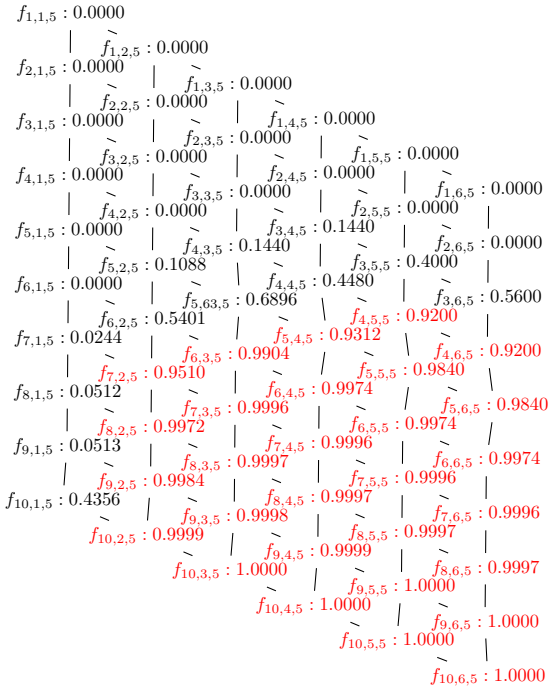

Fig. 2. Upper probabilities $\overline{P}_{f_{\gamma,\delta,\zeta}}(\phi)$ of satisfying the PR ϕ for a subset of the 540 specifications in F over the form $f_{\gamma,\delta,5} = t_\gamma \wedge v_\delta \wedge e_5$, where $\gamma \in \{1,\ldots,10\}$, $\delta \in \{1,\ldots,6\}$. Specifications with $\overline{P}(\phi) \geq 0.9$ in red.

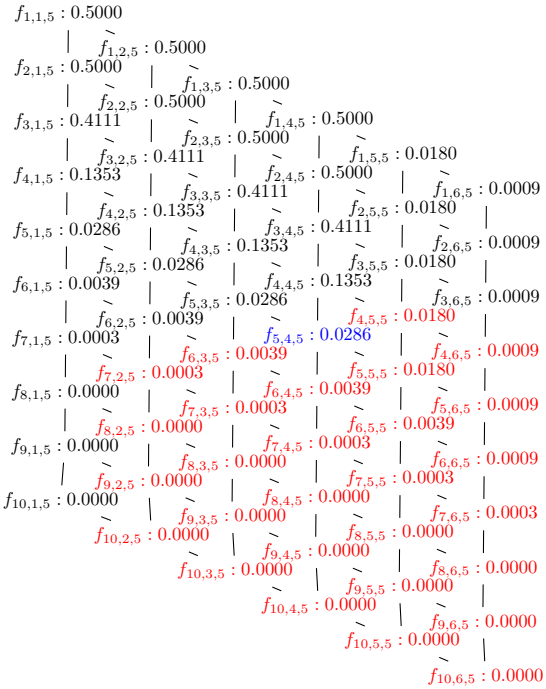

Fig. 3. The membership values for χ for the same subset of specifications as shown Fig. 2, using the 'Sigmoid' function as μ_{χ_1}, and the 'Very Fast' function as μ_{χ_2}. Specifications with $\overline{P}_f(\phi) \geq 0.9$ are shown in red and blue. The specification satisfying $\arg\max\{\mu_\chi(f) : f \in W\}$ within the subset is shown in blue.

Adopting the 'Sigmoid' function for μ_{χ_1} and the 'Very Fast' function for μ_{χ_2} we have that

$$\arg\max\{\mu_\chi(f) : f \in W\} = \{f_{5,4,5}, f_{5,4,4}\},$$

which conform to the VRs to degree $\mu_\chi(f) = 0.0286$. When using the 'Medium' function for μ_{χ_2} so that we have a larger maximum servicing time threshold and a lower maximum velocity threshold than above, then

$$\arg\max\{\mu_\chi(f) : f \in W\} = \{f_{5,4,4}, f_{6,3,4}, f_{6,4,4}\}$$

with $\mu_\chi(f) = 0.8176$.

These "optimal" specifications trade-off an increased risk of collision and a slower service time so as to meet the PR ϕ. The designer can also see the impact of the energy margins, where thresholds above 5 have an identical effect on the probabilities.

For the remaining combinations of membership functions, the computed "optimal" specifications were: when using the 'Sigmoid' function for μ_{χ_1} and the 'Fast' function for μ_{χ_2}

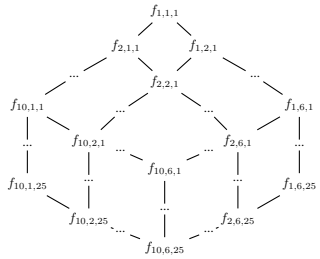

Fig. 4. Hasse diagram of the specifications in the case study, ordered according to their VR compliance from highest (top) to lowest (bottom), servicing time limits on the right side, and velocity limits on the right side.

and when using the 'Linear' function for μ_{χ_1} and the 'Fast' function for μ_{χ_2} we have that, with $\mu_\chi(f) = 0.5$,

$$\arg\max\{\mu_\chi(f) : f \in W\} = \{f_{5,4,5}, f_{5,4,4}\}.$$

If we use the 'Linear' function for μ_{χ_1} and the 'Very Fast'

function for μ_{χ_2} then, with $\mu_\chi(f) = 0.1353$,

$$\arg\max\{\mu_\chi(f) : f \in W\} = \{f_{4,5,4}, f_{4,5,5}\}.$$

Finally, when we use the 'Linear' function for μ_{χ_1} and the 'Medium' function for μ_{χ_2} then, with $\mu_\chi(f) = 0.8176$,

$$\arg\max\{\mu_\chi(f) : f \in W\} = \{f_{5,4,4}, f_{6,3,4}, f_{6,4,4}\}.$$

The differences between choosing a robot that needs to service the human urgently or not, i.e. using the 'Fast' or 'Very Fast', versus the 'Medium' function for μ_{χ_2}, reflect the designer judgements, as the maximum service time threshold is allowed to increase. The desired urgency also influences the velocity threshold, since increasing the maximum velocity limit decreases the servicing time, at a higher collision risk.

VII. Conclusions and Future Work

In this work, we have considered the challenge of exploring different system designs in the presence of multiple requirements of different forms. We proposed a formalization of levels of VR satisfaction using fuzzy logic, to characterize and rank different design options which complements traditional PR formal analysis. The complementary use of these analyses enhances the designers' understanding of their systems, and of the different design options available to them.

We have used a home care assistant case study to illustrate our approach, exploring variations of design constraints.We used the probabilistic model checker PRISM to evaluate the upper probability of PRs holding over a formal model of each system design acting in its environment. We analysed the VRs through a fuzzy logic formalization. A partially ordered set of specifications was generated, and designs that satisfy PRs and VRs to the highest levels possible were identified.

Our technique does not prescribe an algorithm for interpreting or searching the designs, leaving that to the designer. However the number of design candidates can increase exponentially with model detail, similarly to the state explosion problem of model checking. The level of detail required is again up to the designer, and in many circumstances it may be that the expense of analysing the large number of designs is outweighed by the potentially huge cost of adopting an erroneous design, e.g. harm or financial loss.

Future work will apply our approach to more complex designs for which we will need to investigate more computationally efficient methods of generating and exploring the ordered set of specifications. For example, we could analyse only some specifications with respect to PR satisfaction (with model checking), to pre-select a set of specifications for VR analysis, reducing the partially ordered set size.

Acknowledgement.: This work was supported by the EPSRC grants EP/J01205X/1 RIVERAS: Robust Integrated Verification of Autonomous Systems, and EP/K006320/1 and EP/K006223/1: Trustworthy Robotic Assistants.

References

[1] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. CAV*, 2011, pp. 585–591.

[2] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Trans. Auton. Adapt. Syst.*, vol. 4, no. 2, pp. 14:1–14:42, May 2009.

[3] B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, G. Marzo Serugendo, S. Dustdar, A. Finkelstein, C. Gacek, K. Geihs, V. Grassi, G. Karsai, H. M. Kienle, J. Kramer, M. Litoiu, S. Malek, R. Mirandola, H. A. Müller, S. Park, M. Shaw, M. Tichy, M. Tivoli, D. Weyns, and J. Whittle, "Software engineering for self-adaptive systems," B. H. Cheng, R. Lemos, H. Giese, P. Inverardi, and J. Magee, Eds., 2009, ch. Software Engineering for Self-Adaptive Systems: A Research Roadmap, pp. 1–26.

[4] M. Webster, C. Dixon, M. Fisher, M. Salem, J. Saunders, K. L. Koay, K. Dautenhahn, and J. Saez-Pons, "Toward reliable autonomous robotic assistants through formal verification: A case study," *IEEE Transactions on Human-Machine Systems*, no. 99, pp. 1–11, 2015.

[5] M. Gario, A. Cimatti, C. Mattarei, S. Tonetta, and K. Y. Rozier, "Model checking at scale: Automated air traffic control design space exploration," in *Proc. CAV*, 2016, pp. 3–22.

[6] D. Araiza-Illan, D. Western, A. G. Pipe, and K. Eder, "Systematic and Realistic Testing in Simulation of Control Code for Robots in Collaborative Human-Robot Interactions," in *Towards Autonomous Robotic Systems*, 2016, pp. 20–32.

[7] G. Tamura, N. M. Villegas, H. A. Müller, J. P. Sousa, B. Becker, G. Karsai, S. Mankovskii, M. Pezzè, W. Schäfer, L. Tahvildari, and K. Wong, *Software Engineering for Self-Adaptive Systems II: International Seminar*. Springer Berlin Heidelberg, 2013, ch. Towards Practical Runtime Verification and Validation of Self-Adaptive Software Systems, pp. 108–132.

[8] R. Calinescu, C. Ghezzi, M. Kwiatkowska, and R. Mirandola, "Self-adaptive software needs quantitative verification at runtime," *Commun. ACM*, vol. 55, no. 9, pp. 69–77, Sep. 2012.

[9] V. Forejt, M. Kwiatkowska, and D. Parker, "Pareto curves for probabilistic model checking," in *Proc. ATVAS*, 2012.

[10] H. Kress-Gazit, G. Fainekos, and G. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[11] J. Tumova, L. Castro, S. Karaman, E. Frazzoli, and D. Rus, "Minimum-violation LTL planning with conflicting specifications," in *Proc. ACC*, 2013, pp. 200–205.

[12] R. Ehlers and U. Topcu, "Resilience to intermittent assumption violations in reactive synthesis," in *Proc. HSCC*, 2014.

[13] M. Zuluaga, A. Krause, P. Milder, and M. Püschel, ""Smart" design space sampling to predict Pareto-optimal solutions," in *Proc. LCTES*, 2012, pp. 119–128.

[14] E. Letier and A. van Lamsweerde, "Reasoning about partial goal satisfaction for requirements and design engineering," in *Proc. SIGSOFT/FSE*, 2004, pp. 53–62.

[15] M. Serrano, M. Serrano, and J. Sampaio do Prado Leite, "Dealing with softgoals at runtime: A fuzzy logic approach," in *REatRunTime*, 2011.

[16] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, and J. M. Bruel, "RELAX: Incorporating uncertainty into the specification of self-adaptive systems," in *Proc. RE*, 2009, pp. 79–88.

[17] J. Whittle, P. Sawyer, N. Bencomo, B. H. C. Cheng, and J.-M. Bruel, "RELAX: A language to address uncertainty in self-adaptive systems requirement," *Requir. Eng.*, vol. 15, no. 2, pp. 177–196, 2010.

[18] I. J. Jureta, A. Borgida, N. A. Ernst, and J. Mylopoulos, "The requirements problem for adaptive systems," *ACM Trans. Manage. Inf. Syst.*, vol. 5, no. 3, pp. 17:1–17:33, 2014.

[19] B. Hansson, Hansand Jonsson, "A logic for reasoning about time and reliability," *Formal Aspects of Computing*, vol. 6, no. 5, pp. 512–535, 1994.

[20] F. Seo and M. Sakawa, "Fuzzy multiattribute utility analysis for collective choice," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 45–53, 1985.